

DATA COMMUNICATION SYSTEM, DATA COMMUNICATION METHOD
AND DATA COMMUNICATION APPARATUS

BACKGROUND OF THE INVENTION

5 Field of the Invention

The present invention relates to a data communication system, a data communication method and a data communication apparatus, and, more particularly, it relates to a network in which communication is effected at a high speed while mixing information data (including image data) and command data, and a communication protocol applicable to such a network.

Related Background Art

10 In the past, among peripheral equipments for a personal computer (referred to as "PC" hereinafter), hard discs and printers have been used most widely. Such a peripheral equipment has been connected to the PC via a multi-purpose digital interface such as an exclusive I/O interface or an SCSI (small computer system interface).

15 On the other hand, recently, AV (Audio/Visual) equipments such as digital cameras, digital video cameras or the like have also been noticed as one of the peripheral equipment for the PC. Such an AV (Audio/Visual) equipment has also been connected to the PC via an exclusive interface.

25 However, in the conventional communication system,

09288038-040899
668040-8E088260

the digital interface of the peripheral equipment and the digital interface of the AV equipment have no interchangeability, so that these digital interfaces could not interconnected directly. Thus, for example, when a still image is desired to be communicated, the data had to be passed through the PC without fail.

Further, in the conventional exclusive interface and/or SCSI interface, particularly when large capacity data such as a moving image or a still image of the AV equipment is processed, there arose various problems that a data transfer rate becomes low, that a fat communication cable is required for parallel communication, that the number and kind of peripheral equipments capable of being connected are limited, that a connection system is limited and that real time data transfer cannot be effected.

As one of next generation high speed and high performance digital interfaces for solving the above problems, IEEE (The Institute of Electrical and Electronics Engineers, Inc.) 1394-1995 Standard is already known.

The digital interface based upon the IEEE 1394-1995 Standard (referred to as "1394 interface" hereinafter) has the following features:

- (1) A data transfer speed is fast.
- (2) A real time data transfer system (i.e., isochronous transfer system) and an asynchronous

09288038 "040899

(3) A connection construction (topology) having high degree of freedom can be fabricated.

(4) A plug-and-play function and a hot-line insertion/withdrawal function are supported.

However, in the IEEE 1394-1995 Standard, although physical and electrical constructions of connectors and two fundamental data transfer systems are defined, there was no definition regarding how to transmit and receive what kind of data through what kind of data format on the basis of what kind of communication protocol.

Further, in the isochronous transfer system based upon the IEEE 1394-1995 Standard, since response to outgoing packets is not stipulated, it is not ensured whether each isochronous packet is positively received. Accordingly, when it is desired that continuous plural data are positively transferred or when it is desired that one file data is positively transferred while dividing it into plural data, the isochronous transfer system could not be used.

Furthermore, in the isochronous transfer system based upon the IEEE 1394-1995 Standard, even when there is vacancy in transfer band, the total number of transmissions is limited to sixty-four. Thus, when it is desired that many transmissions are effected with smaller transfer bands, the isochronous transfer system

could not be used.

In addition, in the IEEE 1394-1995 Standard, if bus reset is generated in response to ON/OFF of a power source for nodes or connection/disconnection of the node, data transfer must be interrupted. However, in the IEEE 1394-1995 Standard, if the data transfer is interrupted due to the bus reset or error in transmission, it could not be known what kind of data contents are lost. Further, in order to restore the interrupted transfer, very complicated transmission sequence was required. Incidentally, the bus reset refers to a function for automatically effecting recognition of new topology and the setting of address (node ID) assigned to each node. In the IEEE 1394-1995 Standard, this function can provide the plug-and-play function and the hot-line insertion/withdrawal function.

Further, in the communication system based upon the IEEE 1394-1995 Standard, communication protocol in which (although real time ability is not required) object data (for example, still image, graphic data, text data, file data, program data and the like) having relatively large data amount and requiring reliability are continuously transferred while dividing such data into one or more segment data was not proposed concretely.

Particularly, in the communication system based

upon the IEEE 1394-1995 Standard, communication
protocol in which data transmission between plural
equipments is achieved by using a communication system
for broadcasting data in an asynchronous transferring
5 was not also proposed concretely.

Further, when a plurality of such protocols are
proposed, it is desirable that a recipient protocol is
automatically discriminated and is used properly.
However, a method in which a plurality of different
10 communication protocols are used properly was not also
proposed concretely.

SUMMARY OF THE INVENTION

An object of the present invention is to solve the
15 above-described problems.

Another object of the present invention is to
provide a technique in which object data not requiring
real time ability can be transferred positively in a
data communication system, a data communication method
20 and a data communication apparatus.

Also, another object of the present invention is
to provide a technique capable of selecting a usable
communication protocol even when plural different
communication protocols exist in the data communication
25 system, data communication method and data
communication system.

As a preferred embodiment for such objects, the

09288038-040899
668070-DE088260

present invention discloses a data communication system comprising a source node for transferring object data including one or more segments by using at least one asynchronous communication, one or more destination
5 nodes for receiving the object data transferred from the source node, and a controller for setting a logical connection relationship between the source node and the one or more destination nodes and wherein the controller selects a communication protocol used in the
10 source node and the destination nodes among a plurality of different communication protocols.

As another embodiment, the present invention discloses a data communication method comprising the steps of setting a logical connection relationship
15 between a source node and one or more destination nodes, selecting a communication protocol to be used in the source node and the destination nodes among a plurality of different communication protocols, transferring object data including one or more segments
20 to the destination nodes by using at least one asynchronous communication, and receiving the object data transferred from the source node by using the logical connection relationship.

As a further embodiment, the present invention
25 discloses a data communication apparatus comprising a unit for packetizing object data including one or more segments to at least one communication packet, and a

unit for transferring the communication packet in an asynchronous manner by using a logical connection relationship set with respect to one or more destination nodes and a communication protocol.

5 As a still further embodiment, the present invention discloses a data communication method comprising the steps of packetizing object data including one or more segments to at least one communication packet, and transferring the
10 communication packet in an asynchronous manner by using a logical connection relationship set with respect to one or more destination nodes.

 As a further embodiment, the present invention discloses a data communication apparatus comprising a
15 unit for receiving at least one communication packet transferred in an asynchronous manner, by using a logical connection relationship set with respect to one or more destination nodes, and a unit for writing data included in the communication packet into a memory
20 space designated by the communication packet.

 As a still further embodiment, the present invention discloses a data communication method comprising the steps of receiving at least one communication packet transferred in an asynchronous
25 manner, by using a logical connection relationship set with respect to one or more destination nodes, and writing data included in the communication packet into

00288038.040899

a memory space designated by the communication packet.

As a further embodiment, the present invention discloses a data communication apparatus comprising a means for setting a logical connection relationship
5 between a source node and one or more destination nodes, and a means for selecting a communication protocol capable of being used in the logical connection relationship among a plurality of different communication protocols.

10 As the other embodiment, the present invention discloses a data communication method comprising the steps of setting a logical connection relationship between a source node and one or more destination nodes, and selecting a communication protocol capable
15 of being used in the logical connection relationship among a plurality of different communication protocols.

Still other objects of the present invention, and the advantages thereof, will become fully apparent from the following detailed description of the embodiments.

20

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing a communication system according to a preferred embodiment of the present invention;

25 Fig. 2 is a view for explaining a 1394 interface according to a preferred embodiment of the present invention;

Fig. 3 is a conceptional view for explaining a fundamental construction of first and second communication protocols according to a preferred embodiment of the present invention;

5 Figs. 4A and 4B are views for explaining an address space of each node;

Figs. 5A and 5B are views for explaining an output registers (oMCR and oCCR) of a source node according to a preferred embodiment of the present invention;

10 Figs. 6A and 6B are views for explaining an input registers (iMCR and iCCR) of a destination node according to a preferred embodiment of the present invention;

15 Fig. 7 is a view showing command for reading out contents of the oMCR of Fig. 2 and iMCR of Fig. 3;

--- Fig. 8 is a view showing command for writing a predetermined value in the oCCR of Fig. 2 and iCCR of Fig. 3;

20 Fig. 9 is a sequence chart for explaining a procedure till a communication protocol used between the source node and the destination node is set;

Figs. 10A, 10B and 10C are sequence charts for explaining a first communication protocol according to a preferred embodiment of the present invention;

25 Fig. 11 is a view for explaining a procedure for transferring object data by the first communication protocol;

09283038-040899

Figs. 12A and 12B are views for showing a data packet used in the first communication protocol; and

Fig. 13 is a sequence chart for explaining a second communication protocol according to a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The preferred embodiments of the present invention will now be described in detail hereinafter with reference to the accompanying drawings.

Fig. 1 is a view showing an example of a construction of a data communication system according to an embodiment of the present invention. As shown in Fig. 1, the data communication system according to this embodiment is constituted by a computer 10, a camera integrating digital video recorder 28, and a printer 60.

First of all, a construction of the computer 10 will be described. The computer 10 includes a calculation processing unit (MPU) 12 for controlling an operation of the computer 10, a 1394 interface 14 having a function based upon the IEEE 1394-1995 Standard and a function regarding a communication protocol stipulated in this embodiment, an operation unit 16 comprised of a keyboard and a mouse, a detector 18 decoding compressed and coded digital data (moving image data, still image data, voice data and the like),

09283038 040899

09233033 "040899

a display unit (display) 20 comprised of a display device such as a CRT display or a liquid crystal panel, a hard disc 22 for storing various digital data (moving image data, still image data, voice data, graphic data, text data, program data and the like), an internal memory 24, and an internal bus 26 for interconnecting various processing units within the computer 10.

Next, a construction of the camera integrating digital video recorder (referred to as "DVCR" hereinafter) 28 will be described. The DVCR includes an imaging unit (opt) 30 for converting an optical image of an object into an electric signal and for supplying the electric signal to an analogue/digital (A/D) converter 32, an image processing unit 34 for converting digitalized moving image and still image into digital image data having predetermined format, a compression/expansion processing unit 36 having a function for decoding the compressed and coded digital data (moving image data, still image data, voice data and the like) and a function for coding the digital image data with high efficiency (for example, as is in an MPEG system and a DV system, for quantizing and coding the data with variable length after orthogonally converting the data into predetermined image units), a memory 38 for temporarily storing the digital image data coded with high efficiency, a memory 40 for temporarily storing the digital image data not coded

with high efficiency, a data selector 42, a 1394
interface 44 having a function based upon the IEEE
1394-1995 Standard and a function regarding a
communication protocol stipulated in this embodiment, a
5 memory control units 46, 48 for controlling writing and
reading of the memories 38, 40, a control unit (system
controller) 50 adapted to control an operation of the
DVCR 28 and having a microcomputer, an operation unit
52 comprised of a remote controller and an operation
10 panel, an electronic view finder (EVF) 54, a D/A
converter 56, and a recorder/reproducer unit 58
comprised of a recording medium such as a magnetic
tape, a magnetic disc, a photo-magnetic disc or the
like and adapted to record and reproduce various data
15 (moving image data, still image data, voice data and
the like).

Next, a construction of the printer 60 will be
described. The printer includes a 1394 interface 62
having a function based upon the IEEE 1394-1995
20 Standard and a function regarding a communication
protocol stipulated in this embodiment, a data selector
64, an operation unit 66 comprised of an operation
button and a touch panel, a printer controller 68 for
controlling an operation of the printer 60, a decoder
25 70, an internal memory 72, an image processing unit 74
for processing the still image data, text data, graphic
data and the like received through the 1394 interface,

09288038-040899

5

10

15

20

25

The MPU 12 serves to execute software recorded in the hard disc 22 and to shift various data to the internal memory 24. Further, the MPU 12 also serves to

adjust the various processing units connected to the MPU via the internal bus 26.

668040" 3E83260
The 1394 interface 14 serves to receive the image data transferred onto the 1394 serial bus and to send
5 the image data stored in the hard disc 22 or the internal memory 24 to the 1394 serial bus. Further, the 1394 interface 14 serves to transmit the command data for remotely controlling other nodes on the 1394 serial bus. In addition, the 1394 interface 14 has a
10 function for transferring a signal transferred through the 1394 serial bus to other node.

The user or operator can select desired software via the operation unit 16 and cause the MPU 12 to execute the software stored in the hard disc 22.
15 Information regarding the software is displayed to the user through the display unit 20. The decoder 18 serves to decode the image data received from the 1394 serial bus, on the basis of the software. The decoded image data is represented to the user through the
20 display unit 20.

Next, functions and operations of various processing units constituting the DVCR 28 will be described.

In the illustrated embodiment, for example, the
25 DVCR 28 acts as an image transmitting device (source node) for transferring the image data in an asynchronous transferring on the basis of the

communication protocol according to the illustrated embodiment.

5 The imaging (image pickup) unit 30 serves to convert the optical image of the object into the electric signal comprised of a luminance signal (Y) and a color difference signal (C) and to supply the electric signal to the A/D converter 32. The A/D converter 32 serves to digitalize the electric signal.

10 The image processing unit 34 serves to effect predetermined image processing with respect to the digitalized luminance signal and color difference signal and to multiply the signals. The compression/expansion processing unit 36 may process the luminance signal and the color difference signal in
15 parallel by using independent compression processing circuits. Further, they may be processed in a time-share manner by using a common compression processing circuit.

20 Further, in the compression/expansion processing unit 36, in order to increase resistance to transmission path error, the compressed image data is subjected to shuffling process. As a result, continuous code error (i.e., burst error) can be converted into scattered error (i.e., random error)
25 which can easily be corrected or interpolated. When it is desired to make offset of information amount due to roughness/density in the image uniform, before the

092803040899

compressing process, this process is effected. It is advantageous particularly when the coding with variable length such as run length is used.

5 In the compression/expansion processing unit 36, data discriminating information (ID) for restoring the shuffling is added to the compressed image data.

Further, the compression/expansion processing unit 36 adds an error correction code (ECC) to the compressed image data in order to reduce error on
10 recording/reproducing.

The image data compressed in the compression/expansion processing unit 36 is supplied to the memory 38 and the recorder/reproducer unit 58. The recorder/reproducer unit 58 serves to record the added
15 compressed image data such as ID or ECC on a recording medium such as a magnetic tape. The compressed image data is recorded on an independent recording area different from the voice data.

On the other hand, the image data supplied from
20 the image processing unit 34 to the D/A converter 56 is D/A-converted. The EVF 54 serves to display an analogue image signal supplied from the D/A converter 56. Further, the image data processed in the image processing unit 34 is also supplied to the memory 40.
25 Non-compressed image data is stored in the memory 40.

The data selector 42 selects the memory 38 or the memory 40 on the basis of the user's command and

09280033 040899

supplies the compressed image data or the non-compressed image data to the 1394 interface 44.

Further, the data selector 42 supplies the image data supplied from the 1394 interface 44 to the memory 38 or
5 the memory 40.

The 1394 interface 44 serves to transfer the compressed image data or the non-compressed image data in asynchronous transferring on the basis of a communication protocol according to the illustrated
10 embodiment which will be described later. Further, the 1394 interface 44 serves to receive control command for controlling the DVCR 28 through the 1394 serial bus. The received control command is supplied to the control unit 50 through the data selector 42. The 1394
15 interface 44 sends back response to the control command.

Next, functions and operations of various processing units constituting the printer 60 will be described.

20 In the illustrated embodiment, for example, the printer 60 acts as an image receiving device (destination node) for receiving the image data transferred in the asynchronous transferring and for printing the image data, on the basis of the
25 communication protocol according to the illustrated embodiment.

The 1394 interface 62 serves to receive the image

09288038-040899

5 The received image data is supplied to the decoder
70 through the data selector 64. The decoder 70 serves
to decode the image data and output a result to the
image processing unit 74. The image processing unit 74
causes the memory 72 to temporarily store the decoded
0 image data.

On the other hand, the received control command is inputted to the printer controller 68 through the data selector 64. The printer controller 68 performs various controls regarding the printing on the basis of the control data. For example, the printer controller controls sheet feed through the driver 76 and positioning of the printer head 78.

The 1394 interface is functionally constituted by

a plurality of layers. In Fig. 2, the 1394 interface is connected to the 1394 interface of other node via a communication cable 201 based upon the IEEE 1394-1995 Standard. Further, the 1394 interface has one or more communication ports 202, and each communication port is connected to a physical layer 203 included in a hardware portion.

In Fig. 2, the hardware portion includes the physical layer 203 and a link layer 204. The physical layer 203 acts as a physical and electrical interface for other node and serves to effect detection of bus reset and processing therefor, coding/decoding of input and output signals and adjustment of bus usage right. The link layer 204 serves to effect formation of the communication packets, transmission and reception of various communication packets and control of a cycle timer. Further, the link layer 204 provides a function for performing formation and transmission/reception of a data packet used in a second communication protocol which will be described later.

In Fig. 2, a firmware portion includes a transaction layer 205 and a serial bus management 206. The transaction layer 205 controls an asynchronous transfer system and provides various transactions (read, write and lock). Further, the transaction layer 205 provides functions of the first and second communication protocols which will be described later.

09288038 1040899 668040 " 8E088260

The serial bus management 206 provides functions for effecting control of its node, management of connection condition of its node, management of ID information of its node, resources management of the serial bus network, on the basis of an IEEE 1212 CSR Standard which will be described later.

The hardware portion and firmware portion shown in Fig. 2 substantially constitute the 1394 interface, and fundamental constructions thereof are stipulated in the IEEE 1394-1995 Standard.

An application layer 207 included in the software portion differs from each other depending upon application software used and controls how to transfer what kind of object data by using the first and second communication protocols.

The first and second communication protocols (described later) serves to expand the functions of the hardware portion and the firmware portion which constitute the 1394 interface and to provide a new transfer sequence to the software portion.

Next, fundamental constructions of the first and second communication protocols according to the illustrated embodiment will be described with reference to Fig. 3.

In Fig. 3, the reference numeral 300 denotes a controller; 302 denotes a source node; 304 denotes n ($n \geq 1$) (number) destination nodes; 306 denotes a subunit

09288038-040889

of the source node; and 308 denotes object data such as moving image data, still image data, graphic data, text data, file data, program data or the like.

5 The reference numeral 310 denotes a first memory space within the destination node 304 and is designated by predetermined destination offset (#0); and 312 denotes first connection showing a logical relationship (i.e., connection) between the source node 302 and the destination node 304. The destination offset means
10 address for commonly designating the memory spaces of N (number) destination nodes 304.

 The reference numeral 314 denotes n-th memory space within the destination node 304 and is designated by predetermined destination offset (#n); and 316
15 denotes n-th connection showing a logical connection relationship (i.e., connection) between the source node 302 and the destination node 304.

 In the illustrated embodiment, each node controls or governs the first memory space 310 to n-th memory
20 space 314 on the basis of 64-bit address space based upon the IEEE 1212 CSR (Control and Status Register Architecture) Standard (or ISO/IEC 13213:1994 Standard). The IEEE 1212 CSR Standard is Standard for stipulating control, management and address assignment
25 of the serial bus.

 Figs. 4A and 4B are views for explaining the address space of each node. Fig. 4A shows address

09288038 040899

space of 64-bit for addressing the address space of each node. Upper 16 bits of the 64-bit address is used to designate a certain node in certain bus. Fig. 4B shows a portion of the address space shown in Fig. 4A, and, for example, used to designate a certain address space of the node designated by the upper 16 bits. The first memory space 310 to n-th memory space 314 shown in Fig. 3 are designated by 48-bit address shown in Fig. 4B (referred to as "destination offset" hereinafter). In Fig. 4B, for example, 000000000000_{16} to $0000000003FF_{16}$ are reserves areas, and, areas where the object data 308 is actually written are areas where the lower 48 bits of the address become $FFFF0000400_{16}$ and so on.

15 In Fig. 3, the source node 302 means a node having a function for transferring the object data 308 in accordance with the first or second communication protocol, and the destination node 304 means a node for having a function for receiving the object data 308 transferred from the source node 302. Further, the controller 300 means a node having a function for setting and managing connection (i.e., logical connection relationship) between the source node 302 and the one or mode destination nodes 304 and having a function for selecting the communication protocol used in such connection.

The controller 300, source node 302 and

destination node 304 may be operated in independent nodes. Alternatively, the controller 300 and the source node 302 may be operated in a single same node. Further, the controller 300 and the destination node 304 may be operated in a single same node. In this case, the transaction between the controller 300 and the source node 302 or the destination node 304 can be omitted, thereby simplifying the communication sequence.

10 In the illustrated embodiment, a case where the controller 300, source node 302 and destination node 304 are operated in independent nodes will be explained. For example, the computer 10 having the 1394 interface 14 acts as the computer 300. Further, 15 the DVCR 28 having the 1394 interface 14 acts as the source node 302 and the printer 60 having the 1394 interface 14 acts as the destination node 304.

In the illustrated embodiment, as shown in Fig. 3, one or more connections can be set between the source node 302 and one or more destination node 304. When it is required that certain object data be transferred, each connection is set by the controller 300 on the basis of the output register (Figs. 5A and 5B) of the source node 302 and input register (Figs. 6A and 6B) of the destination node 304 which will be described later.

Incidentally, in the illustrated embodiment, the computer 10 as the controller 300 may be operated as

the destination node 304. In this case, the connections are set between one source node 302 and two destination nodes 304 to transfer the object data 308.

Further, in the illustrated embodiment, while an example that the computer 10 is operated as the controller 300 was explained, it is not necessary that the computer 10 acts as the controller 300. The DVCR 28 or the printer 60 may be operated as the controller 300.

Next, the register space of the source node 302 and the register space of the destination node 304 according to the illustrated embodiment will be explained with reference to Figs. 5A, 5B, 6A and 6B. Figs. 5A and 5B are views each explaining a construction of the output register of the source node 302. Figs. 6A and 6B are views each explaining a construction of the input register of the destination node 304.

Each register space is prepared in a predetermined area on the address space shown in Figs. 4A and 4B, and, when the controller 300 sets the connection between the source node 302 and the destination node 304, it is selected which first or second communication protocol is assigned to the connection.

Fig. 5A shows an output master connection register (referred to as "oMCR" hereinafter). The oMCR is mainly a register space (or memory space) describing

communication ability of the source node 302.

Uppermost field 501 (2 bits) of the oMCR is a max data rate field indicating a maximum output bit rate of the source node 302. An example of values set in the max data rate field is shown in the following Table 1.

A next field 502 (1 bit) is a flow control flag indicating whether a first communication protocol (i.e., protocol using broadcast sending) according to the illustrated embodiment) can be used or not. For example, when the value of this field is "1", it indicates that the first communication protocol can be used. If the value is "0", it indicates that the second communication protocol can be used. A next field 503 (5 bits) is a reserved field which is prepared in the future.

Table 1

max data rate	Maximum bit rate
00 ₂	100 Mbps
01 ₂	200 Mbps
10 ₂	400 Mbps
11 ₂	reserved

A next field 504 (6 bits) is an extension field which is used when an output FIFO size field (described later) is expanded in the future. A next field 505 (10 bits) is an output FIFO size field indicating a memory

capacity of a pre-incoming/pre-outgoing buffer memory (referred to as "FIFO memory" hereinafter) used for outputting data.

A next field 506 (3 bits) is a reserved field which is prepared in the future. A lowermost field 507 (5 bits) is a number of CCRs field indicating the number of output connection control registers (Fig. 5B) (described later), i.e., the number of connections which can be set in the source node 302.

Fig. 5B shows an output connection control register (referred to as "oCCR" hereinafter). The oCCR is a register space (or memory space) describing as to one of connections set in the source node 302. The source node 302 may have a single oCCR or plural oCCRs.

In the oCCR, an uppermost field 510 (2 bits) is a data rate field indicating output bit rate used in the connection set in the oCCR. An example of values set in the data rate field is shown in "Table 2".

Table 2

data rate	bit rate
00_2	100 Mbps
01_2	200 Mbps
10_2	400 Mbps
11_2	reserved

A next field 511 (1 bit) is a broadcast flag field

indicating which communication protocol is used in the connection set in the oCCR. When a value of the broadcast flag field is "1", it indicates that the first communication protocol is used, for example.

5 A next field 512 (1 bit) is an on-line field indicating whether the oCCR is used or not. When a value of the on-line field is "1", it indicates that the oCCR is used.

 A next field 513 (18 bits) is a reserved field
10 which is prepared in the future. A next field 514 (10 bits) is a payload size field indicating a data amount of a data packet transferred by one data transferring. The data amount shows a payload size of the data packet and does not include additional data such as packet
15 header.

 A next field 515 (16 bits) is a connection ID
 field in which discrimination information (i.e., connection ID) for discriminating the connection set in the oCCR. A last field 516 (48 bits) is an offset
20 address field indicating a top address of the predetermined memory space reserved in the destination node 304.

 Fig. 6A shows an input master connection register (referred to as "iMCR" hereinafter). The iMCR mainly
25 is a register space (or memory space) describing communication ability of the destination node 304.

 An uppermost field 601 (2 bits) of the iMCR is a

09288038 040899 668040" BE088260

max data rate field indicating a maximum bit rate of the destination node 304. Values set in the max data rate may be the values shown in the above Table 1, for example.

5 A next field 602 (1 bit) is a flow control flag field indicating whether the first communication protocol (i.e., protocol using broadcast sending according to the illustrated embodiment) can be used or not. For example, when the value of this field is "1",
10 it indicates that the first communication protocol can be used. If the value is "0", it indicates that the second communication protocol can be used. A next field 603 (5 bits) is a reserved field which is prepared in the future.

15 A next field 604 (6 bits) is an extension field which is used when a maximum receive buffer size field (described later) is expanded in the future. A next field 605 (10 bits) is a maximum receive buffer size field indicating a maximum capacity of a receiving
20 buffer used for inputting data. The maximum receive buffer size field indicates a maximum data amount of one segment.

 A next field 606 (3 bits) is a reserved field which is prepared in the future. A lowermost field 607
25 (5 bits) is a number of CCRs field indicating the number of input connection control registers (Fig. 6B) (described later), i.e., the number of connections

0928038-040899

which can be set in the destination node 304.

Fig. 6B shows an input connection control register (referred to as "iCCR" hereinafter). The iCCR is a register space (or memory space) describing one of connections set in the destination node 304. The destination node 304 may have only one iCCR or may have a plurality of iCCRs.

In the iCCR, an uppermost field 610 (2 bits) is a data rate field indicating input bit rate used in the connection set in the iCCR. Values set in the data rate field may be the values in the above Table 2, for example.

A next field 611 (1 bit) is a broadcast flag field indicating which communication protocol is used in the connection set in the iCCR. When a value of the broadcast flag field is "1", it indicates that the first communication protocol is used, for example.

A next field 612 (1 bit) is an on-line field indicating whether the iCCR is used or not. When a value of the on-line field is "1", it indicates that the iCCR is used. A next field 613 (18 bits) is a reserved field which is prepared in the future.

A next field 614 (10 bits) is a receive buffer size field indicating a size of the memory space reserved in the destination node 304, i.e., a data amount of one segment.

A next field 615 (16 bits) is a connection ID

field in which discrimination information (i.e., connection ID) for discriminating the connection set in the iCCR.

5 A last field 616 (48 bits) is an offset address field indicating a top address of the predetermined memory space reserved in the destination node 304.

10 Next, command for reading out the contents of the registers shown in Figs. 5A, 5B, 6A and 6B and command for writing predetermined values in the registers will be explained.

15 Each command is packetized in an asynchronous packet based upon the IEEE 1394-1995 Standard and is transferred to the designated node. As shown in Fig. 7, the asynchronous packet is a data packet having unit of 1 Quadlet (4 bytes = 32 bits).

The controller 300 according to the illustrated embodiment can read out the value in the oMCR shown in Fig. 5A and iMCR shown in Fig. 6A by using READ MCR status command shown in Fig. 7.

20 In Fig. 7, a field 701 (16 bits) is a destination ID field for designating node ID of the recipient. A field 702 (6 bits) is a transaction level (tl) field for designating tag inherent to each transaction.

25 A field 703 (2 bits) is a retry (rt) code for designating whether the packet effects retry or not. A field 704 (4 bits) is a transaction code (tcode). The tcode designates format of the packet and a type of

668040" 8E088260

transaction which must be executed. In the illustrated embodiment, a value of this field is selected to "0001₂", for example, and requests for a process (i.e., write transaction) for writing the data field of this packet in the memory space of the destination offset field 707. Further, when the value is selected to "0005₂", a reading process (i.e., read transaction) is requested.

5 A field 705 (4 bits) shows priority (pri) and designates a preferential order. In the illustrated embodiment, a value of this field is selected to "0000₂". A field 706 (16 bits) is a source ID field for designating node ID of the recipient.

10 A field 707 (48 bits) is a destination offset field for designating the address space of the node designated by the destination ID field 701. A field 708 (16 bits) is a data length field for indicating a length a data field (described later) as a unit of byte.

15 A field 709 (16 bits) is an extended tcode field. In the illustrated embodiment, a value of this field is selected to "0000₂". A field 710 (32 bits) a header CRC field in which error detecting codes for detecting errors in the fields 701 to 709.

20 Next, a construction of data fields will be explained. In the illustrated embodiment, if a data amount stored in the data field does not reach the

0928038.040899
668040" 8E08260

multiple of "Quadlet", "0" is added until the multiple of "Quadlet" is obtained.

5 A field 711 (4 bits) is a common/transaction set (CTS) field. In the illustrated embodiment, since the CTS defined by AV/C Digital Interface Command Set General Specification Standard (referred to as "AV/C Command Standard" hereinafter) is used a value "0₁₆" is set in this field.

10 A field 712 (4 bits) is a ctype/response field for designating various commands and responses to such command. For example, when status command is designated, a value "1₁₆" is set in this field. When response corresponding to this status command (i.e., status response) is designated, a value "C₁₆" is set in
15 this field.

20 A field 713 (5 bits) is a subunit type field for designating an imaginary function unit (i.e., subunit) of a certain node. The "subunit" is defined in the AV/C Command Standard and, for example, includes VTR subunit, printer subunit or the like.

A field 714 (3 bits) is a subunit ID field. If the function units designated by the subunit type field 713 are more than 1 (one), this field designates ID numbers for discriminating such function units.

25 A field 715 (8 bits) is an opcode field for designating the kind of command. When the READ MCR command according to the illustrated embodiment is

669040" 8E0B2250

designated, for example, a value "06₁₆" is set in this field.

Fields 716 (8 bits) to 719 (8 bits) are operand[0] to operand[3] fields, contents of which are
5 differentiated in dependence upon the contents of the ctype/response field 712 and the opcode field 715. For example, in case of the READ MCR status command, a value "FF₁₆" is set in each field. On the other hand, in case of the READ MCR status response, a value of the
10 oMCR of Fig. 5A or a value of the iMCR of Fig. 6B is set.

A value "0" is set in a field 720 (8 bits). A field 721 (32 bits) is a data CRC field in which error detecting codes for detecting errors in the fields 711 to 720 are stored, as is in the header CRC field.
15

The controller 300 according to the illustrated embodiment can write a predetermined value in the oCCR of Fig. 5B or the iCCR of Fig. 6B by using SET CCR control command shown in Fig. 8.

20 In Fig. 8, fields 701 to 710 included in a packet header are constituted similar to Fig. 7. Accordingly, hereinafter, constructions of data fields of SET CCR control command will be described.

A field 801 (4 bits) is a CTS field in which a
25 value "0₁₆" is set, as is in the READ MCR status command.

A field 802 (4 bits) is a ctype/response field for

5580140 "BEBB260

designating various commands and responses to such command. For example, when control command is designated, a value "0₁₆" is set in this field. When response indicating receipt of this control command is returned, a value "9₁₆" indicating ACCEPTED response is set in this field. On the other hand, when response indicating the fact that the control command is not received is returned, a value "A₁₆" indicating REJECTED response is set in this field.

10 A field 803 (5 bits) is a subunit type field for designating a predetermined subunit, as is in the READ MCR status command of Fig. 7. A field 804 (3 bits) is a subunit ID field for designating an ID number for discriminating a predetermined subunit.

15 A field 805 (8 bits) is an opcode field for designating the kind of command. When the SET CCR command according to the illustrated embodiment is designated, for example, a value "07₁₆" is set in this field.

20 A field 806 (2 bits) is a data rate field for indicating input bit rate used by the connection set in the desired oCCR (or iCCR). Values set in the data rate field are the same as the values shown in the above Table 2, for example.

25 A field 807 (1 bit) is a broadcast flag field including a flag for indicating which communication protocol is used by the connection set in the desired

092880310-040899
668040-81088260

A field 808 (1 bit) is an on-line field. A value "1" is set in this field to request the use of the desired oCCR (or iCCR). If the on-line field 512 (or 612) of the desired oCCR (or iCCR) is "1", since the oCCR (or iCCR) has already been used, the node which received the SET CCR control command returns the REJECTED response.

15 A field 812 (8 bits) becomes a payload size field when the oCCR is set and a receive buffer field when the iCCR is set. An amount of data which can be transferred by one data packet is set in the payload size field. Further, a data amount of one segment is set in the receive buffer field.

A field 815 (48 bits) is an offset address field for setting top address of the predetermined memory

space reserved in the destination node 304. A field 816 (8 bits) is a CCR number field in which a number for designating the desired oCCR (or iCCR) among one or plural oCCRs (or iCCRs).

5 A field 817 (32 bits) is a data CRC field for storing error detecting codes for detecting errors in the fields 801 to 816.

10 Next, a procedure till the controller 300 sets the communication protocol used between the source node 302 and the destination node 304 will be described with reference to Fig. 9. In Fig. 9, the controller 300, source node 302 and destination node 304 effect negotiation (described hereinbelow) to set the communication protocol to be used.

15 First of all, the controller 300 sends the READ MCR status command packet shown in Fig. 7 to the source node 302 (901). The source node 302 returns the response packet in which the values of the fields 501 to 507 of the oMCR of the source node to the controller 20 300 (901).

 Then, the controller 300 sends the READ MCR status command packet shown in Fig. 7 to the destination node 304 (902). The destination node 304 returns the response packet in which the values of the fields 601 25 to 607 of the iMCR of the destination node to the controller 300 (902).

 By the above operation, the controller 300 which

09283038-1040899

read out the oMCR of the source node 302 and the iMCR of the destination node 304 judges main communication ability of each node from the contents of each register and selects the available communication protocol.

5 For example, if the flow control flag 502 of the oMCR and the flow control flag 602 of the iMCR are both "1", the controller 300 judges that the source node 302 and the destination node 304 can correspond to the first communication protocol (i.e., protocol by which
10 the broadcast data transferring is effected) which will be described later. In this case, the controller 300 selects the use of the first communication protocol.

 After selection of the first communication protocol, the controller 300 produces the SET CCR
15 control command shown in Fig. 8 in order to set the logical connection relationship between the source node 302 and the destination node 304.

 In the SET CCR control command for setting the oCCR of the source node 302 (or the iCCR of the
20 destination node 304), the controller 300 sets the value "1" in the broadcast flag field 807, the value "1" in the on-line field 808, the available connection IDs managed by the controller 300 in the connection ID fields 812, 813, the available destination offset of
25 the destination node 304 in the offset address field 814, and the CCR number designating non-used oCCR (or iCCR) in the CCR number field 815, respectively.

09288038 040899 668040 " 88088260

Further, if either or both of the flow control flag 502 of the oMCR and the flow control flag 602 of the iMCR is "0", the controller 300 judges that the source node 302 and the destination node 304 cannot correspond to the first communication protocol (i.e., protocol by which the broadcast data transferring is effected) which will be described later. In this case, the controller 300 selects the use of the second communication protocol which will be described later.

After selection of the second communication protocol, the controller 300 produces the SET CCR control command shown in Fig. 8 in order to set the logical connection relationship between the source node 302 and the destination node 304.

In the SET CCR control command for setting the oCCR of the source node 302; the controller 300 sets the value "0" in the broadcast flag field 807, the value "1" in the on-line field 808, and the data amount which can be transferred by the single data packet in the payload size field 811, respectively.

Further, in the SET CCR control command for setting the iCCR of the destination node 304, the controller 300 sets the data amount of one segment in the receive buffer size field 811.

Further, in the SET CCR control command for setting the oCCR (or iCCR), the controller 300 sets the available destination offset of the destination node

09288038-040899
668040"8E88260

304 in the offset address field 814 and sets the CCR number for designating the non-used oCCR (or iCCR) in the CCR number field 815.

5 In the illustrated embodiment, the value set in the receive buffer size field by the controller 300 is the integral multiple of the value set in the payload size field. With this arrangement, the data transferring efficiency can be improved.

668040" 3E88260
10 Then, the controller 300 sends the above-mentioned SET CCR control command to the source node 302 in order to set the oCCR (903). The SET CCR control command includes the value indicating the communication protocol selected by the controller 300 and values required for the communication protocol. If the
15 condition of the source node 302 can correspond to the SET CCR control command, the source node 302 returns the ACCEPTED response to the controller 300 (903). Thereafter, the source node 302 starts the data transferring on the basis of the selected communication
20 protocol.

On the other hand, if the condition of the source node 302 cannot correspond to the SET CCR control command, the source node 302 returns the REJECTED response to the controller 300. In this case, the data
25 transferring is not started, the logical connection is released, and the nodes start negotiation from initiation again.

Then, the controller 300 sends the above-mentioned SET CCR control command to the destination node 304 in order to set the iCCR (904). The SET CCR control command includes the value indicating the communication
5 protocol selected by the controller 300 and values required for the communication protocol.

If the condition of the destination node 304 can correspond to the SET CCR control command, the destination node 304 returns the ACCEPTED response to
10 the controller 300 (904). Thereafter, the destination node 304 starts the data transferring on the basis of the selected communication protocol.

On the other hand, if the condition of the destination node 304 cannot correspond to the SET CCR
15 control command, the destination node 304 returns the REJECTED response to the controller 300. In this case, the data transferring is not started, the logical connection is released, and the nodes start negotiation from initiation again.

By the negotiation having the above-mentioned procedure, the controller 300 can set the connection between the source node 302 and the destination node 304, and the communication protocol used in the
20 connection. Incidentally, in Fig. 9, while an example that one destination node 304 is used was explained,
25 when there are a plurality of destination nodes, the above-mentioned processing may be performed for each

0928038-040899

destination node. In this case, the controller 300 selects the available communication protocol in accordance with the communication abilities of all of the nodes in question.

5 Next, the data transferring based upon the first communication protocol according to the illustrated embodiment will be described.

 The first communication protocol serves to realize multi-cast communication by utilizing the broadcast
10 communication.

 In the first communication protocol, one object data is divided into one or more segments, and each segment is packetized to one or more "asynchronous broadcast packets", and each asynchronous broadcast
15 packet is broadcast-transferred by "asynchronous broadcast transaction". Here, the asynchronous broadcast packet and the asynchronous broadcast transaction are new packet format and transfer method defined in the first communication protocol according
20 to the illustrated embodiment.

 Now, the transferring procedure based upon the first communication protocol will be fully described with reference to Figs. 10A and 10C. Fig. 10A is a sequence chart for explaining the transferring
25 procedure or sequence between one source node and one destination node, and Fig. 10C is a sequence chart for explaining the transferring procedure or sequence

0928038-1040899
6680401-8808260

5

0

5

0

5

Now, an example of the transferring procedure for the object data 308 will be described. In Fig. 11, the object data 308 is, for example, still image data having a data size of 128 Kbytes.

5 The source node 302 divides the object data 308 into one or more segments in accordance with the receiving capacity of each destination node 304 recognized by using the send request packet. The data size of one segment is variably set by the source node 10 302 on the basis of the size of the internal buffer of each destination node 304. For example, as shown in Fig. 11, if the internal buffer having the same size as that of the object data 308 can be reserved in each destination node 304, the source node 302 divides the 15 object data 308 into one segment.

Then, the source node 302 divides each segment into one or more segment data. Fig. 11 shows an example that one segment is divided into 500 segment data (one segment data = 256 bytes).

20 The source node 302 successively broadcasts the segments by using at least one asynchronous broadcast transaction. Fig. 11 shows an example that one segment data is transferred by using one asynchronous broadcast transaction.

25 After all of the segment data of one segment are transferred, if there is next segment, the source node 302 prepares for transferring of the next segment;

0928038-040899
668040 "BEBB250

whereas, if there is next segment, the communication to the destination node 304 is terminated (1008, 1009 in Figs. 10A and 10C).

5 Next, an operation of the controller 300 will be fully described with reference to Figs. 10A and 10C.

The controller 300 transfers a transaction command packet to the source node 302 in an asynchronous manner (1001 in Figs. 10A and 10C).

10 The source node 302 which received the transaction command packet effects the initial setting by using the connection ID notified from the controller 300 (1002, 1003 in Figs. 10A and 10C) and executes the asynchronous broadcast transaction (1004 to 1007 in Figs. 10A and 10C). By this asynchronous broadcast
15 transaction, the source node 302 can transfer one or more segments of the object data 308 successively.

20 Incidentally, in the first communication protocol according to the illustrated embodiment, the controller 300 provides a function for controlling setting/releasing of the connection. Accordingly, the transferring of the object data 308 after the setting of the connection is carried out by the negotiation between the source node 302 and the destination node 304.

25 After a series of asynchronous broadcast transactions are finished, the source node 302 broadcasts an asynchronous broadcast packet indicating

05-28-03-10-14-08-99

segment end (referred to as "segment end packet" hereinafter) (1008 in Figs. 10A and 10C).

After the segment end packet is received from the source node 302, the controller 300 releases the connection to terminate the data transferring.

Since the segment end packet is broadcasted, the contents of this packet can be detected also in the destination node 304. Accordingly, in place of the controller 300, the destination node 304 can be constituted to release the connection to the source node 302 by the destination node itself.

Next, an operation of the source node 302 will be fully described with reference to Figs. 10A and 10C.

The source node 302 which received the transaction command packet from the controller 300 sends the above-mentioned send request packet to each destination node 304 (1002 in Figs. 10A and 10C).

The send request packet is a packet for obtaining required initial information before the asynchronous broadcast transaction for the object data 308 is started. The connection ID designated by the controller 300 is written in this packet.

The destination node 304 broadcasts the ack response packet which is response to the send request packet (1003 in Figs. 10A and 10C). The same connection ID as that of the send request packet is stored in the ack response packet. Accordingly, the

09288038-040899

5 The sizes of the internal buffers reserved in the destination nodes 304 are stored in the ack response packet. After the ack response packet is received, the source node 302 starts the asynchronous broadcast transaction by using the destination offset commonly designating the memory spaces of the destination nodes 304. The destination offset is address designated by the controller 300 during the negotiation.

After the first asynchronous broadcast packet is sent, the source node 302 is waiting for the response from the destination node 304. The response packet storing the connection ID and the sequence number is sent from the destination node 304 as a form of the asynchronous broadcast packet. After the response packet is received, the source node 302 effects increment of the sequence number and transfer the asynchronous broadcast packet including the next segment data (1005 in Figs. 10A and 10C).

By repeating such procedure, the source node 302 executes the asynchronous broadcast transactions successively (1006, 1007 in Figs. 10A and 10C). A maximum time period for waiting for the response from the destination node 304 is previously determined, and, if the response is not received after the maximum time period is elapsed, the same data is sent again by using the same sequence number.

Further, when a response packet requesting the re-sending is transferred from the destination node 304, the source node 302 can broadcast the data having the designated sequence number.

After all of the segments of the object data 308 were transferred by using the asynchronous broadcast transactions, the source node 302 broadcasts the segment end packet to terminate the data transferring (1008 in Figs. 10A and 10C).

Incidentally, in the above-mentioned embodiment, while an example that the response packet is sent without fail in the asynchronous broadcast transaction of one segment data was explained, the present invention is not limited to such an example. After the data buffer of the destination node 304 is filled with a plurality of continuous segment data, the destination node 304 may send the response packet.

Next, an operation of the destination node 304 will be fully described with reference to Figs. 10A and

05283038 040859

5

10

15

20

25

of the destination node, the destination node 304 discards such receiving packet.

Further, if difference in the sequence number of the receiving packet is detected, the destination node 304 can send a response packet indication re-send request. In this case, the destination node 304 informs the source node 302 of the sequence number for the re-send request.

After all of the asynchronous broadcast transactions are finished, the segment end packet is sent from the source node 302. When this packet is received, the destination node 304 terminates the data transferring process (1008 in Figs. 10A and 10C).

After the segment end packet is received, the destination node 304 broadcasts a response packet indicating the fact that the segment end packet is properly received (1009 in Figs. 10A and 10C).

As mentioned above, the communication system according to the illustrated embodiment can eliminate inconvenience of the conventional communication systems. Further, in the data transferring not requiring the real time feature, the data can easily be transferred at a high speed.

Further, in the illustrated embodiment, after the connection is set by the controller 300, the transfer processing of the object data is executed between the source node 302 and each destination node 304 without

0522033 040399

control of the controller 300. As a result, a simple communication protocol in which the load of the controller 300 is reduced and complicated communication procedure is not required can be provided.

5 Furthermore, in the illustrated embodiment, the destination node 304 is designed to return the response to each asynchronous broadcast transaction. With this arrangement, a communication protocol capable of positively transferring the data not requiring the real
10 time ability can be provided.

 In order to realize more positive data transferring, if the data transferring is interrupted due to bus reset or any transfer error, it is necessary that the data transferring is quickly re-started
15 without losing any data. Now, a re-starting procedure defined by the communication protocol according to the illustrated embodiment will be described with reference to Fig. 10B.

 For example, if the bus reset occurs after the
20 asynchronous broadcast packet having the sequence number i was received, each node interrupts the transferring process and executes initialization of the bus, confirmation of the connection relationship and setting of the node ID, in accordance with the
25 procedures defined in the IEEE 1394-1995 Standard (1010, 1011 in Fig. 10B).

 After re-construction of the bus is completed,

each destination node 304 broadcasts a resend request packet in which the connection ID and the sequence number i are stored (1012 in Fig. 10B).

5 If the asynchronous broadcast transaction can be re-started, the source node 302 confirms the connection ID of the received resend request packet and broadcasts the ack response packet in which the connection ID is stored (1013 in Fig. 10B).

10 Thereafter, the source node 302 broadcasts the segment data having sequence numbers next to the sequence number requested by the resend request packet, i.e., segment data starting from the sequence number ($i + 1$) successively (1014 in Fig. 10B).

15 By the above-mentioned procedure, the controller 300, source node 302 and destination node 304 can re-start the data transferring easily and positively without considering their node IDs, even if the data transferring is interrupted.

20 Further, as mentioned above, in the illustrated embodiment, even if the data transferring is interrupted, a control procedure of the controller 300 can be simplified.

25 Next, a construction of the asynchronous broadcast packet used in the first communication protocol will be explained with reference to Fig. 12A. The asynchronous broadcast packet is a data packet having unit of 1 Quadlet (4 bytes = 32 bits), for example.

05288038040399

First of all, an arrangement of a packet header 1219 will be described.

In Fig. 12A, a field 1201 (16 bits) indicates the destination ID and node ID of the recipient (i.e., destination node 304). In the communication protocol according to the illustrated embodiment, in order to realize the asynchronous broadcast transaction of the object data 308, a value of this field is selected for broadcast ID (i.e., "FFFF₁₆").

A field 1202 (6 bits) is a transaction level (tl) field which is a tag inherent to the transaction.

A field 1203 (2 bits) is a retry (rt) code field for designating whether the packet effects retry or not.

A field 1204 (4 bits) indicates a transaction code (tcode). The tcode designates format of the packet and type of transaction to be executed. In the illustrated embodiment, a value of this field is selected to "0001₂", for example, and a process (i.e., write transaction) for writing a data block 522 of this packet in the memory field of a destination offset field 1207 is requested.

A field 1205 (4 bits) is a priority (pri) field, and, in the illustrated embodiment, a value of this field is selected to "0000₂".

A field 1206 (16 bits) is a source ID field for indicating node ID of the sender (i.e., source node

09288038 1040899

302).

A field 1207 (48 bits) is the destination offset field for commonly designating lower 48 bits of the address spaces of the destination nodes 304.

5 A field 1208 (16 bits) is a data length field indicating a length of a data field (described later) in byte unit.

10 A field 1209 (16 bits) is an extended tcode field. In the illustrated embodiment, a value of this field is selected to "0000₂".

A field 1210 (32 bits) is a header CRC field in which error detecting codes for detecting error in the fields 1201 to 1209.

15 Next, a construction of a data block 1220 will be described. The data block 1220 includes header (packet) information 1221 and a data field 1222.

20 Connection ID for discriminating logic connection relationships (connections) between the nodes is stored in the header information 1221. Incidentally, a construction of the header information 1221 is varied in accordance with purpose of usage.

25 The data field 1222 is a field having variable length in which the segment data are stored. If the number of the segment data stored in the data field is not multiple of Quadlet, "0" is added until the Quadlet is reached.

A field 1211 (16 bits) is a connection ID field in

09288033 040899

which the connection ID according to the illustrated
embodiment is stored. The 1394 interface according to
the illustrated embodiment discriminates the
connections set between the source node 302 and one or
5 more destination nodes 304 on the basis of the
connection ID stored in this field. In the illustrated
embodiment, $2^{16} \times (\text{node number})$ connections can be
established. Thus, a plurality of connections can be
set until a total amount of communication bands used in
10 each connection reach the capacity of the transfer
path.

A field 1212 (8 bits) is a protocol type field
indicating a communication procedure (i.e., kind of
communication protocol) based upon the header
15 information 1221. When indicating the communication
protocol according to the illustrated embodiment, a
value of this field becomes " 01_{16} ", for example.

A field 1213 (8 bits) shows control flags, in
which predetermined control data for controlling a
20 communication procedure of the communication protocol
according to the illustrated embodiment and the like
are set. In the illustrated embodiment, an uppermost
bit of this field is selected to a resend request flag,
for example. Accordingly, when a value of the
25 uppermost bit of this field becomes "1", it is
indicated that resend request based upon the
communication protocol according to the illustrated

663040" 8E0B8260

embodiment occurs.

5 A field 1214 (16 bits) is a sequence number field
in which continuous values (i.e., sequence numbers) are
set regarding the packets transferred on the basis of
the particular connection ID (i.e., connection ID
designated in the field 511). By using such sequence
numbers, the destination node 304 can monitor
continuity of the segment data transferred successively
by the asynchronous broadcast transaction. If there is
10 no coincidence, the destination node 304 can effect
resend request on the basis of the sequence number.

15 A field 1215 (16 bits) is a reconfirmation number
field. In the illustrated embodiment, this field is
used only if the value of the resend request flag is
"1". For example, if the value of the resend request
flag is "1", the sequence number of the resend request
packet is set in this field.

20 A field 1216 (16 bits) is a buffer size field.
The buffer size of the destination node 304 is set in
this field.

A field 1217 (16 bits) is a reserved field which
is prepared for the future.

25 A field 1218 (32 bits) is a data CRC field in
which error detecting codes for detecting errors in the
fields 1211 to 1217 are stored, as is in the header CRC
field.

Incidentally, in the first communication protocol

00000000000000000000000000000000

according to the illustrated embodiment, while an example that the asynchronous broadcast transaction is realized by using the asynchronous broadcast packet shown in Fig. 12A was explained. The present invention is not limited to such an example. For example, the asynchronous broadcast transaction can be realized by using an asynchronous stream packet shown in Fig. 12B. In this case, the object data is divided into one or more segments, and each segment is packetized to one or more asynchronous stream packets. The source node 302 broadcast-transferring the asynchronous stream packets successively within an asynchronous transferring period. A format and a transferring method for the asynchronous stream packet are defined in IEEE 1394.a Standard which is an extended Standard of the IEEE 1394-1995 Standard.

In Fig. 12B, a field 1240 (16 bits) is a data length field. A length of a data field 1246 (described later) is stored as byte unit in this field. A field 1241 (2 bits) is a tag field in which a value "00₂" is stored.

A field 1242 (6 bits) is a channel field. A channel number assigned to a series of the asynchronous stream packets including its packet are set in this field. A field 1243 (4 bits) is a transaction code (tcode) field for designating a value "A₁₆" indicating the asynchronous stream packet.

A field 1244 (4 bits) is an asynchronous code (sy) field. Control code determined in dependence upon the application used is stored in this field. A field 1245 (32 bits) is a header CRC field in which error
5 detecting codes for detecting errors in the fields 1241 to 1244 are stored.

A field (variable length) 1246 is a data field in which values of the fields 1211 to 1217 shown in Fig. 12A and the segment data are stored. If the segment
10 data stored in the field 1246 is not the multiple of Quadlet, "0" is added until the Quadlet is reached. A field 1247 (32 bits) is a data CRC field. Similar to the header CRC field, an error detecting code for detecting an error in the field 1246 is this field.

15 Next, the second communication protocol according to the illustrated embodiment will be described.

The second communication protocol is a communication protocol utilizing a communication system in which the recipient is identified and realizing
20 unicast communication.

In the second communication protocol, one object data is divided into one or more segments, and each segment is packetized to "asynchronous write request packets" and is transferred by "asynchronous write
25 transaction". The asynchronous write request packet and the asynchronous write transaction are a packet format and a transfer method defined in the IEEE 1394-

09288038 040899 668040 8E08B260

Now, a transferring procedure based upon the second communication protocol will be fully described with reference to Fig. 13.

After the above-mentioned negotiation (Fig. 9) is finished, first of all, the controller 300 sends receive segment command to the destination node 304 (1301). As soon as a receiving condition is prepared, the destination node 304 sends INTERIM response to the controller 300 and assumes a waiting condition (1302).

Then, the controller 300 sends send segment command to the source node 302 (1303). The source node 302 sends INTERIM response to the controller 300 and assumes a data receiving condition (1304).

After the data receiving condition is established, the source node 302 sends the desired object data 308 to the destination node 304 by using the asynchronous write transaction (1305-1308).

Now, an example of a transferring procedure for transferring the object data 308 will be explained. In Fig. 11, the object data 308 is still image data having data size of 128 Kbytes.

The source node 302 divides the object data 308 into one or more segments. For example, if the internal buffer having the same size as that of the object data 308 could be reserved in the destination node 304, the source node 302 divides the object data

into one segment.

Then, the source node 302 detects the contents of the payload size field 514 of the oCCR designated by the controller 300 and divides each segment into one or more segment data. Fig. 11 shows an example that the object data 308 of one segment is divided into 500 segment data (one segment = 256 bytes). The contents of the payload size field 514 is set by the controller 300.

Then, the source node 302 transfers the segments successively by using one asynchronous write transaction. Fig. 11 shows an example that one segment data is transferred by using one asynchronous write transaction.

When a total amount of data successively transferred by one or more asynchronous write transaction reaches the value set in the receive buffer size field 614 (i.e., capacity of the internal buffer), the source node 302 sends ACCEPTED response to the send segment command to the controller 300 (1309).

Then, the destination node 304 sends ACCEPTED response to the receive segment command to the controller 300 and informs of the fact that the data transferring of one segment is completed (1310). If there is a next segment, the controller 300, source node 302 and destination node 304 repeat the aforementioned procedures.

09283038 040899

In this way, the source node 302 can positively send the desired object data 308 to the destination node 304 by using the asynchronous write transaction.

As mentioned above, in the above-mentioned
5 embodiments, the logical connection relationship which does not rely upon the physical connection style can be established in the bus-type network such as defined by the IEEE 1394-1995 Standard.

Further, according to the illustrated embodiment,
10 in communication systems based upon the IEEE-1394-1995 Standard, there can be provided a new communication protocol in which object data having a relatively large amount of data and not requiring real time ability but requiring high reliability (for example, still image
15 data, graphic data, text data, file data, program data or the like) is divided into one or more segments and the segments are successively transferred by using the asynchronous transferring.

Further, according to the illustrated embodiment,
20 in communication systems based upon the IEEE 1394-1995 Standard, there can be provided a new communication protocol in which data communication between plural equipments is realized by using broadcast communication of asynchronous transfer type.

25 In addition, according to the illustrated embodiment, in communication systems based upon the IEEE 1394-1995 Standard, even if there are a plurality

09283038-040899

of different communication protocols, the available communication protocol can automatically be discriminated and be used properly.

(Other embodiments)

5 The communication protocols explained in the aforementioned embodiments and various processing operations required for realizing the such communication protocols can also be realized by software.

10 For example, a recording medium storing program code for realizing the function of each embodiment may be supplied to the control portion (for example, MPU 12, system controller 50, printer controller 68 in Fig. 1) of the equipment constituting the communication
15 system according to each embodiment so that the control portion reads out the program code stored in the recording medium and controls the operation of the communication system or the equipment itself to execute the function of each embodiment in accordance with the
20 program code.

 Alternatively, a recording medium storing program code for realizing the function of each embodiment may be supplied to the 1394 interface 14, 44 or 62 of each equipment so that the control portion (for example,
25 serial bus management 206 in Fig. 2) for controlling the operation of the 1394 interface 14, 44 or 62 controls the processing operations to execute the

function of each embodiment in accordance with the program code.

In this case, the program code itself read out from the recording medium realizes the function of each embodiment, and the program code itself and a means (for example, the recording medium itself) for supplying the program code to the control portion form a part of the present invention.

Such a recording medium storing the program code may, for example, a floppy disc, a hard disc, an optical disc, a photo-magnetic disc, CD-ROM, a magnetic tape, a non-volatile memory card, ROM or the like.

It should be noted that a case where the program code read out from the recording medium cooperates with OS (operating system) or one of various application software operating on the control portion to realize the function of each embodiment is also included in the present invention.

Further, it should be noted that a case where, after the program code read out from the recording medium is stored in a memory of a function expanding unit connected to the control portion, a control portion of the function expanding unit executes part or all of the actual processing in accordance with the program code stored in the memory to realize the function of each embodiment is also included in the present invention.

The invention may be embodied in other specific forms without departing from the spirits or essential characteristics thereof.

For example, in the illustrated embodiments, while
5 an example that, when the controller 300 reads out the
oMCR of the source node 302 and the iMCR of the
destination node 304, the controller sends the READ MCR
status command shown in Fig. 7 to the nodes was
explained, the present invention is not limited to such
10 an example. So long as the addresses of the oMCR and
the iMCR may previously be constructed to become
specific addresses, the controller 300 can read out the
contents of the registers by using the asynchronous
read transaction defined in the IEEE 1394-1995
15 Standard.

In such a case, the 1394 interface according to
the illustrated embodiment can realize a part of the
processing of the negotiation shown in Fig. 9 without
affecting an influence upon the layers more upper than
20 the transaction layer shown in Fig. 2. Thus, the 1394
interface according to the illustrated embodiment can
reduce the load on the application layers and
facilitates hardware specification.

Further, in the illustrated embodiments, while an
25 example that, when the controller 300 writes the
predetermined values in the oCCR of the source node 302
and the iCCR of the destination node 304, the

09283038 040859

controller sends the SET CCR control command shown in Fig. 8 to the nodes was explained, the present invention is not limited to such an example. So long as the addresses of the oCCR and the iCCR may previously be constructed to become specific addresses, the controller 300 can write the contents of the registers by using asynchronous read write compare swap lock transaction defined in the IEEE 1394-1995 Standard.

In such a case, the 1394 interface according to the illustrated embodiment can realize a part of the processing of the negotiation shown in Fig. 9 by response to the lock transaction. Thus, since the 1394 interface according to the illustrated embodiment can realize the above-mentioned processing by the response in the transaction layer, the processing speed can be more improved. Further, the load on the application layers can be reduced and hardware specification can be facilitated.

In the illustrated embodiments, while the construction and procedure of the controller 300 capable of selecting two different communication protocols was explained, the controller can select the communication protocol among three or more communication protocols. In this case, each node can indicate a support condition of each communication protocol by the reserved field 503 of the oMCR or the

reserved field 506 of the iMCR.

Further, in the illustrated embodiments, while the communication protocols applicable to the network based upon the IEEE 1394-1995 Standard was explained, the present invention is not limited to such protocols. The communication protocols according to the present invention can be applied to the bus-type network based upon the IEEE 1394-1995 Standard or a network capable of imaginarily constructing such a bus-type network.....

Therefore, the above-mentioned embodiments are merely examples in all respects, and must not be construed to limit the invention.

The scope of the present invention is defined by the scope of the appended claims, and is not limited at all by the specific descriptions of this application. Furthermore, all the modifications and changes belonging to equivalents of the claims are considered to fall within the scope of the present invention.

09288038 040899